R_note

```
#Rstudio: A free, open-source integrated development environment (IDE) for R
#R markdown: enable you to embed R code and R output directly into document, pdf, HTML
```

```
#Assign and print value
x<-1:5
y=6:10
x
```

```
## [1] 1 2 3 4 5
```

```
y
```

```
## [1]  6  7  8  9 10
```

```
print(x)
```

```
## [1] 1 2 3 4 5
```

```
#R is case sensitive(X won't work)
#You can import data set from the click in the left or use read.table()

#Basic functions

#What's stored in the working menu
ls()
```

```
## [1] "x" "y"
```

```
# remove x from working menu
rm(x)

#Arithmetic operation
sqrt(4) #square root of x
```

```
## [1] 2
```

```
log(1000) #Natural logarithm of x
```

```
## [1] 6.907755
```

```
exp(4) # exponential of x , anti-log
```

```
## [1] 54.59815
```

```
log2(4) #log base 2 of x, other number are plausible
```

```
## [1] 2
```

```
abs(-4) #absolute value of x
```

```
## [1] 4
```

```
#Repetitive numbers and sequence
m<-seq(from=1, to=3, by =0.25) #choose the range then set the difference
n<- rep(1:3, times=3) #character and sequance can also be repeated
m+n #every elements will be sum with x
```

```
## [1] 2.00 3.25 4.50 2.75 4.00 5.25 3.50 4.75 6.00
```

```
seq(1:3)+seq(4) #if the elements are not equal, then your can't to the calculation
```

```
## Warning in seq(1:3) + seq(4): 較長的物件長度並非較短物件長度的倍數
```

```
## [1] 2 4 6 5
```

```
n[n<3] # use index to find specific element
```

```
## [1] 1 2 1 2 1 2
```

```
#matrix
matrix(c(1,2,3,4,5,6,7,8,9),nrow=3,byrow = TRUE) #create a matrix - byrow means the order of the numbers are by row
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
```

```
#Create dataframe
?data.frame
```

```
## starting httpd help server ... done
```

```
aa <- data.frame(nickname=c("John","Mary","Leo"), weight=60:62, Height=c(160,170,180))# each parameters has its own elements - every column is a variable - consist of three vectors
names(aa) #find the names of each variable
```

```
## [1] "nickname" "weight"   "Height"
```

```
mean(aa$weight) #use $ to extract value from data
```

```
## [1] 61
```

```
attach(aa) #attach database to R search path - pro: call the variable directly, con: account
 for working memories
mean(weight) #now you don't have to use $ for searching
```

```
## [1] 61
```

```
detach(aa)#just detach the database
levels(aa$weight) #show levels attribute of a variable
```

```
## NULL
```

```
?levels
summary(aa) #show some statistical information
```

```
##    nickname             weight         Height   
##  Length:3           Min.   :60.0   Min.   :160  
##  Class :character   1st Qu.:60.5   1st Qu.:165  
##  Mode  :character   Median :61.0   Median :170  
##                     Mean   :61.0   Mean   :170  
##                     3rd Qu.:61.5   3rd Qu.:175  
##                     Max.   :62.0   Max.   :180
```

```
as.factor(aa$weight) #turn the variable into factor
```

```
## [1] 60 61 62
## Levels: 60 61 62
```

```
dim(aa) #dimension of the data (number of observations, number of variables)
```

```
## [1] 3 3
```

```
length(aa) #=3
```

```
## [1] 3
```

```
aa[1:2,]  #first two rows
```

nickname	weight	Height
<chr>	<int>	<dbl>

	nickname <chr>	weight <int>	Height <dbl>
1	John	60	160
2	Mary	61	170

2 rows

```
#import data
?read.csv #readcsv file - file.choose() allows you to search the file directly , header= whet
her the first row is header
?read.table #read table - sep=how the elements were separated
?read.delim#read tab-delimited file , txt file
#some parameters - header = T(whether there is header), sep=','(seperation)
#or you can just type the import dataset on the right side then find your file

#export data
?write.table #it will be saved at current working directory - this can be saved with many for
mats like csv,txt
?write.csv#save as csv file
#parameters - row.name=T(whether there is row name)

#setting and loading working directory
getwd()#find current working directory
```

```
## [1] "C:/Users/Work/Desktop/Quantitavie Method for Deicsion Making"
```

```
?setwd() #set the working directory #or you can just click session and choose working directo
ry

?save.image()# save as a work space image file #or just click session and save
?load() #input the file name or load(file.choose())- to directly choose the file or just clic
k session

#writing script
#click file to open or create a R script
#Run multiple lines of code - just select them all and run
#type and click tab key - Rstudio will show possible options of commands
```

```
#combining data
bb<-cbind(aa,aa$nickname=='John') #bind the data with another data with additional column
bb
```

nickname <chr>	weight <int>	Height <dbl>	aa$nickname == "John" <lgl>
John	60	160	TRUE
Mary	61	170	FALSE
Leo	62	180	FALSE

3 rows

```
cc<-rbind(bb,c('Lien',52, 171, FALSE)) #bind with additional row
cc
```

nickname <chr>	weight <chr>	Height <chr>	aa$nickname == "John" <chr>
John	60	160	TRUE
Mary	61	170	FALSE
Leo	62	180	FALSE
Lien	52	171	FALSE

4 rows

```
#Apply function
data1<-matrix(seq(1,10),nrow=5,ncol = 2,byrow=F)
data1
```

```
##      [,1] [,2]
## [1,]    1    6
## [2,]    2    7
## [3,]    3    8
## [4,]    4    9
## [5,]    5   10
```

```
?apply #MARGIN-1 means row, 2 means column, FUN - means function
apply(data1,2, mean, na.rm=TRUE) #na.rm - remove missing values
```

```
## [1] 3 8
```

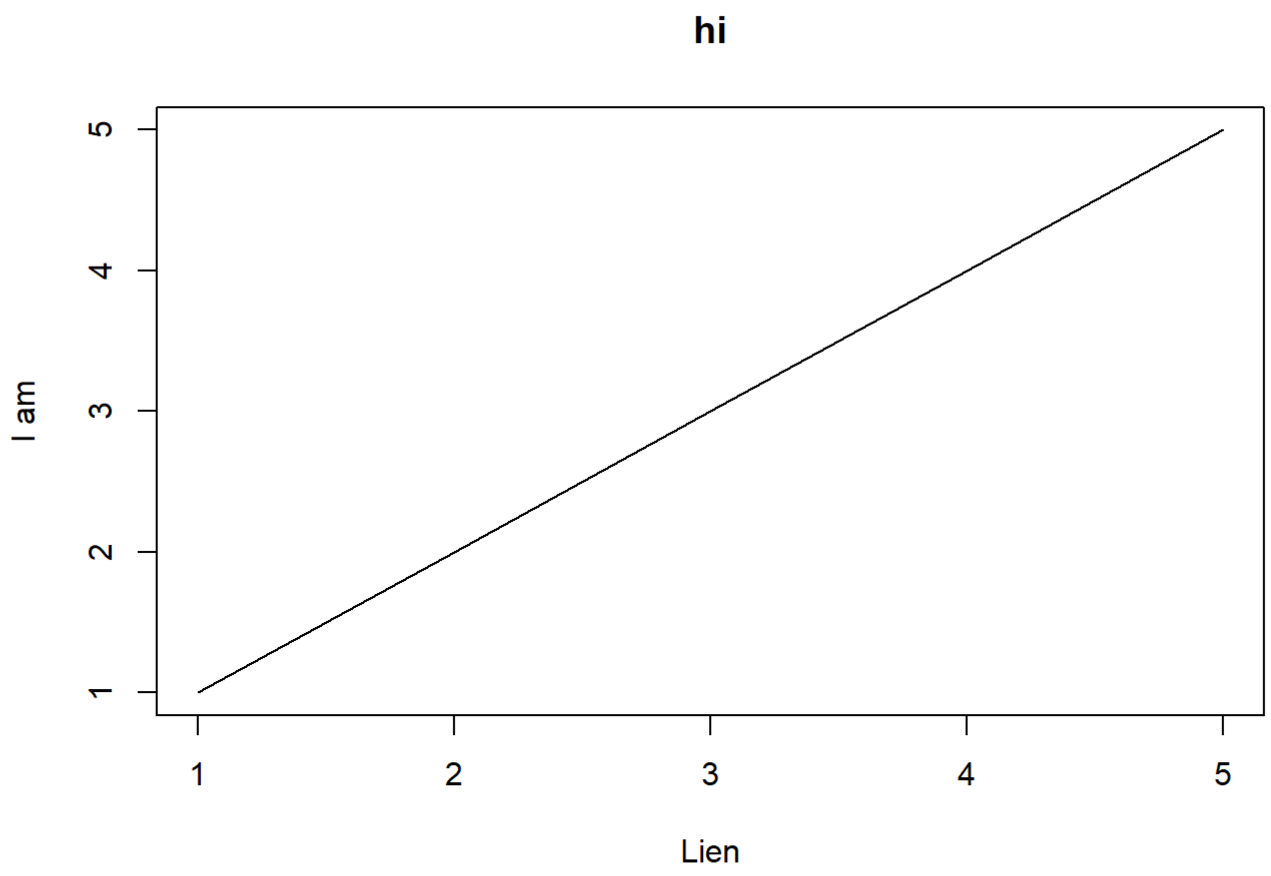
```
colMeans(data1,na.rm=TRUE) #calculate column mean
```

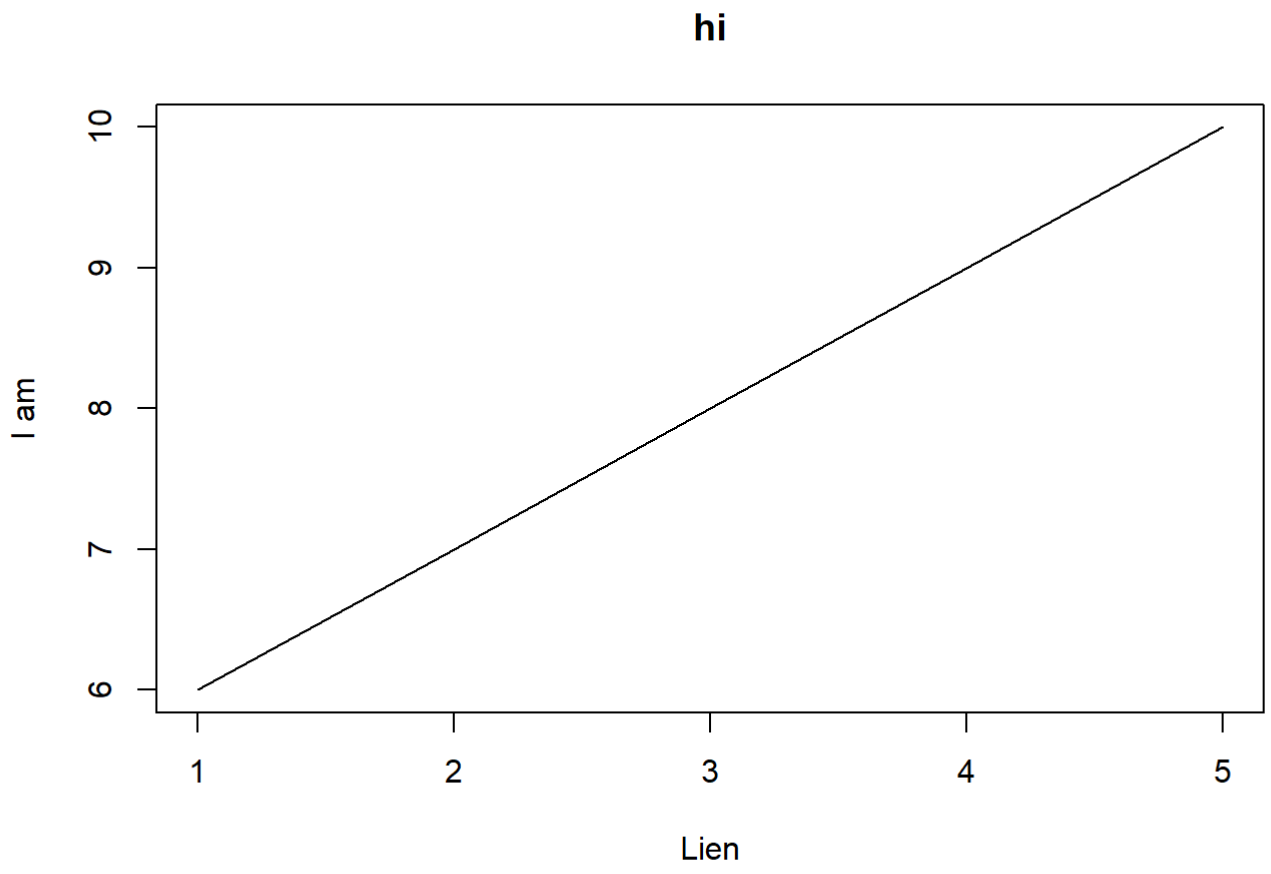
```
## [1] 3 8
```

```
apply(data1,2,quantile,prob=c(0.2,0.8))# quantile-calculate percentile, prob=c(0.2,0.8)- whic
h percentile to calculate
```

```
##     [,1] [,2]
## 20%  1.8  6.8
## 80%  4.2  9.2
```

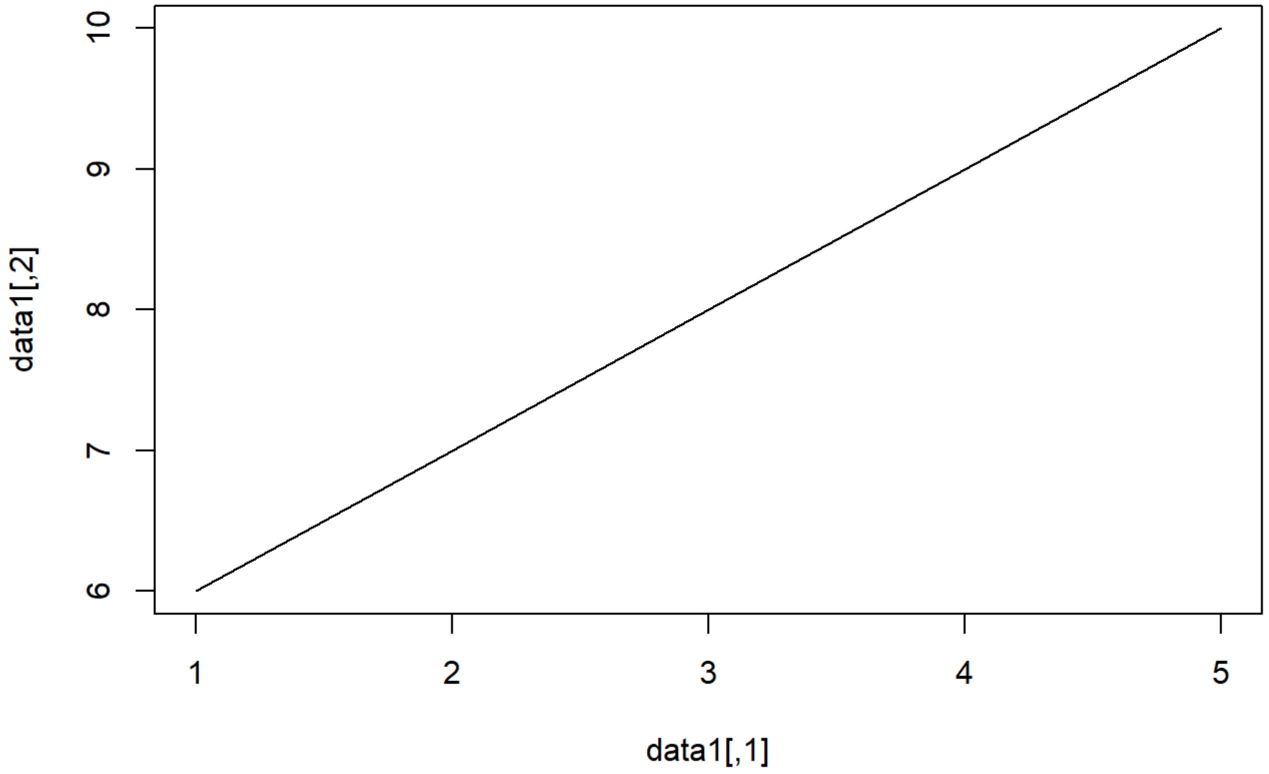
```
apply(data1,2,plot,type='l', main='hi', xlab='Lien', ylab='I am') #use plot function to creat
e a line plot(type='l')
```





```
## NULL
```

```
plot(data1, type='l')
```



```
apply(data1,1,sum) #calculate the sum of the row
```

```
## [1]  7  9 11 13 15
```

```
rowSums(data1,na.rm=T) #same calculate the row sum
```

```
## [1]  7  9 11 13 15
```

```
#tapply function
?tapply # INDEX- a grouping variable to create subsets of data, simplify-let R know to simpli
fy the result
heightdata<-data.frame('age'=c(13,12,14,15,15,13),'height'=c(168,157,167,177,169,172))
heightdata
```

age <dbl>	height <dbl>
13	168
12	157
14	167
15	177

age <dbl>	height <dbl>
15	169
13	172

6 rows

```
attach(heightdata)
tapply(age,height>160,summary) #find subsets to calculate
```

```
## $`FALSE`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##      12      12      12      12      12      12 
## 
## $`TRUE`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##      13      13      14      14      15      15
```

```
#or you can use by command
?by

#install package
?install.packages() #install package , or leave it blank it will let you choose or you can click tools to install
library('base')#load the library of this package to available the commands - disappear after end of R session
help(package=base)
?remove.packages()

rm(list=ls()) #remove all variables from global environment
```